# ARBITRATION SYSTEM HAVING A PACKET MEMORY AND METHOD FOR MEMORY RESPONSES IN A HUB-BASED MEMORY SYSTEM

## TECHNICAL FIELD

This invention relates to computer systems, and, more particularly, to a computer system including a system memory having a memory hub architecture.

## BACKGROUND OF THE INVENTION

Computer systems use memory devices, such as dynamic random access memory ("DRAM") devices, to store data that are accessed by a processor. These memory devices are normally used as system memory in a computer system. In a typical computer system, the processor communicates with the system memory through a processor bus and a memory controller. The processor issues a memory request, which includes a memory command, such as a read command, and an address designating the location from which data or instructions are to be read. The memory controller uses the command and address to generate appropriate command signals as well as row and column addresses, which are applied to the system memory. In response to the commands and addresses, data are transferred between the system memory and the processor. The memory controller is often part of a system controller, which also includes bus bridge circuitry for coupling the processor bus to an expansion bus, such as a PCI bus.

Although the operating speed of memory devices has continuously increased, this increase in operating speed has not kept pace with increases in the operating speed of processors. Even slower has been the increase in operating speed of memory controllers coupling processors to memory devices. The relatively slow speed of memory controllers and memory devices limits the data bandwidth between the processor and the memory devices.

In addition to the limited bandwidth between processors and memory devices, the performance of computer systems is also limited by latency problems that increase the time required to read data from system memory devices. More specifically,

when a memory device read command is coupled to a system memory device, such as a synchronous DRAM ("SDRAM") device, the read data are output from the SDRAM device only after a delay of several clock periods. Therefore, although SDRAM devices can synchronously output burst data at a high data rate; the delay in initially providing the data can significantly slow the operating speed of a computer system using such SDRAM devices.

One approach to alleviating the memory latency problem is to use multiple memory devices coupled to the processor through a memory hub. In a memory hub architecture, a memory hub controller is coupled over a high speed data link to several memory modules. Typically, the memory modules are coupled in a point-to-point or daisy chain architecture such that the memory modules are connected one to another in series. Thus, the memory hub controller is coupled to a first memory module over a first high speed data link, with the first memory module connected to a second memory module through a second high speed data link, and the second memory module coupled to a third memory module through a third high speed data link, and so on in a daisy chain fashion.

Each memory module includes a memory hub that is coupled to the corresponding high speed data links and a number of memory devices on the module, with the memory hubs efficiently routing memory requests and memory responses between the controller and the memory devices over the high speed data links. Each memory requests typically includes a memory command specifying the type of memory access (*e.g.*, a read or a write) called for by the request, a memory address specifying a memory location that is to be accessed, and, in the case of a write memory request, write data. The memory request also normally includes information identifying the memory module that is being accessed, but this can be accomplished by mapping different addresses to different memory modules. A memory response is typically provided only for a read memory request, and typically includes read data as well as an identifying header that allows the memory hub controller to identify the memory request corresponding to the memory response. However, it should be understood that memory requests and memory responses having other characteristics may be used. In any case,

in the following description, memory requests issued by the memory hub controller propagate downstream from one memory hub to another, while memory responses propagate upstream from one memory hub to another until reaching the memory hub controller. Computer systems employing this architecture can have a higher bandwidth because a processor can access one memory device while another memory device is responding to a prior memory access. For example, the processor can output write data to one of the memory devices in the system while another memory device in the system is preparing to provide read data to the processor. Moreover, this architecture also provides for easy expansion of the system memory without concern for degradation in signal quality as more memory modules are added, such as occurs in conventional multi drop bus architectures.

Although computer systems using memory hubs may provide superior performance, they nevertheless may often fail to operate at optimum speeds for a variety of reasons. For example, even though memory hubs can provide computer systems with a greater memory bandwidth, they still suffer from latency problems of the type described above. More specifically, although the processor may communicate with one memory device while another memory device is preparing to transfer data, it is sometimes necessary to receive data from one memory device before the data from another memory device can be used. In the event data must be received from one memory device before data received from another memory device can be used, the latency problem continues to slow the operating speed of such computer systems.

Another factor that can reduce the speed of memory transfers in a memory hub system is the transferring of read data upstream (*i.e.*, back to the memory hub controller) over the high-speed links from one hub to another. Each hub must determine whether to send local responses first or to forward responses from downstream memory hubs first, and the way in which this is done affects the actual latency of a specific response, and more so, the overall latency of the system memory. This determination may be referred to as arbitration, with each hub arbitrating between local requests and upstream data transfers.

There is a need for a system and method for arbitrating data transfers in a system memory having a memory hub architecture to lower the latency of the system memory.

## SUMMARY OF THE INVENTION

According to one aspect of the present invention, a memory hub module includes a decoder that receives memory requests and determines a memory request identifier associated with each memory request. A packet memory receives memory request identifiers and stores the memory request identifiers. A packet tracker receives remote memory responses, associates each remote memory response with a memory request identifier and removes the memory request identifier from the packet memory. A multiplexor receives remote memory responses and local memory responses. The multiplexor selects an output responsive to a control signal. Arbitration control logic coupled to the multiplexor and the packet memory develops the control signal to select a memory response for output.

## BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a block diagram of a computer system including a system memory having a high bandwidth memory hub architecture according to one example of the present invention.

Figure 2 is a functional block diagram illustrating an arbitration system included in the hub controllers of Figure 1 according to one example of the present invention.

Figure 3a and 3b are functional illustrations of a packet memory shown in Figure 2 according to one example of the present invention.

## DETAILED DESCRIPTION OF THE INVENTION

A computer system 100 according to one example of the present invention is shown in Figure 1. The computer system 100 includes a system memory 102 having a memory hub architecture including a plurality of memory modules 130,

each memory module including a corresponding memory hub 140. Each of the memory hubs 140 arbitrates between memory responses from the memory module 130 on which the hub is contained and memory responses from downstream memory modules, and in this way the memory hubs effectively control the latency of respective memory modules in the system memory by controlling how quickly responses are returned to a system controller 110, as will be described in more detail below. In the following description, certain details are set forth to provide a sufficient understanding of the present invention. One skilled in the art will understand, however, that the invention may be practiced without these particular details. In other instances, well-known circuits, control signals, timing protocols, and/or software operations have not been shown in detail or omitted entirely in order to avoid unnecessarily obscuring the present invention.

The computer system 100 includes a processor 104 for performing various computing functions, such as executing specific software to perform specific calculations or tasks. The processor 104 is typically a central processing unit ("CPU") having a processor bus 106 that normally includes an address bus, a control bus, and a data bus. The processor bus 106 is typically coupled to cache memory 108, which, as previously mentioned, is usually static random access memory ("SRAM"). Finally, the processor bus 106 is coupled to the system controller 110, which is also sometimes referred to as a "North Bridge" or "memory controller."

The system controller 110 serves as a communications path to the processor 104 for the memory modules 130 and for a variety of other components. More specifically, the system controller 110 includes a graphics port that is typically coupled to a graphics controller 112, which is, in turn, coupled to a video terminal 114. The system controller 110 is also coupled to one or more input devices 118, such as a keyboard or a mouse, to allow an operator to interface with the computer system 100. Typically, the computer system 100 also includes one or more output devices 120, such as a printer, coupled to the processor 104 through the system controller 110. One or more data storage devices 124 are also typically coupled to the processor 104 through the system controller 110 to allow the processor 104 to store data or retrieve data from

internal or external storage media (not shown). Examples of typical storage devices 124 include hard and floppy disks, tape cassettes, and compact disk read-only memories (CD-ROMs).

The system controller 110 also includes a memory hub controller ("MHC") 132 that is coupled to the system memory 102 including the memory modules 130a,b...n, and operates to apply commands to control and access data in the memory modules. The memory modules 130 are coupled in a point-to-point architecture through respective high speed links 134a and 134b coupled between the memory module 130a and the memory hub controller 132 and between adjacent memory modules 130a-n. The high speed link 134a is the downlink, carrying memory requests from the memory hub controller 132 to the memory modules 130a-n. The high speed link 134b is the uplink, carrying memory responses from the memory modules 130a-n to the memory hub controller 132. The high-speed links 134a and 134b may be optical, RF, or electrical communications paths, or may be some other suitable type of communications paths, as will be appreciated by those skilled in the art. In the event the high-speed links 134a and 134b are implemented as optical communications paths, each optical communication path may be in the form of one or more optical fibers, for example. In such a system, the memory hub controller 132 and the memory modules 130 will each include an optical input/output port or separate input and output ports coupled to the corresponding optical communications paths. Although the memory modules 130 are shown coupled to the memory hub controller 132 in a point-to-point architecture, other topologies that may be used, such as a ring topology, will be apparent to those skilled in the art.

Each of the memory modules 130 includes the memory hub 140 for communicating over the corresponding high-speed links 134a and 134b and for controlling access to eight memory devices 148, which are synchronous dynamic random access memory ("SDRAM") devices in the example of Figure 1. The memory hubs 140 each include input and output ports that are coupled to the corresponding high-speed links 134a and 134b, with the nature and number of ports depending on the characteristics of the high-speed links. A fewer or greater number of memory devices

148 may be used, and memory devices other than SDRAM devices may also be used. The memory hub 140 is coupled to each of the system memory devices 148 through a bus system 150, which normally includes a control bus, an address bus, and a data bus.

As previously mentioned, each of the memory hubs 140 executes an arbitration process that controls the way in which memory responses associated with the memory module 130 containing that hub and memory responses from downstream memory modules are returned to the memory hub controller 132. In the following description, memory responses associated with the particular memory hub 140 and the corresponding memory module 130 will be referred to as "local responses," while memory responses from downstream memory modules will be referred to as "downstream responses." In operation, each memory hub 140 executes a desired arbitration process to control the way in which local and downstream responses are returned to the memory hub controller 132. For example, each hub 140 may give priority to downstream responses and thereby forward such downstream responses upstream prior to local responses that need to be sent upstream. Conversely, each memory hub 140 may give priority to local responses and thereby forward such local responses upstream prior to downstream responses that need to be sent upstream. Examples of arbitration processes that may be executed by the memory hubs 140 will be described in more detail below.

Each memory hub 140 may execute a different arbitration process or all the hubs may execute the same process, with this determination depending on the desired characteristics of the system memory 102. It should be noted that the arbitration process executed by each memory hub 140 is only applied when a conflict exists between local and downstream memory responses. Thus, each memory hub 140 need only execute the corresponding arbitration process when both local and downstream memory responses need to be returned upstream at the same time. Other examples of arbitration schemes are described in application 10/690,810 entitled "Arbitration System and Method for Memory Responses in a Hub-Based Memory System", incorporated herein by reference.

An example of an arbitration system 200 included in the hub controllers 140 of Figure 1 is shown in Figure 2. A downlink receiver 202 receives memory requests. The memory requests include an identifier and a request portion, which includes data in the event the request is a write request. The identifier is referred to herein as a packet ID or a memory request identifier. A decoder 204 is coupled to the downlink receiver 202 and determines the memory request identifier associated with each memory request. The memory request identifiers are stored in a packet memory 206. The packet memory 206 shown in Figure 2 is a first-in, first-out (FIFO) memory, but other buffering schemes may be used in other embodiments. In this manner, a packet ID or memory request identifier associated with each memory request passed to a hub controller is stored in the packet memory 206. When the packet memory 206 is a FIFO memory, the memory request identifiers are stored in time order. In the following description, memory requests associated with the particular memory hub 140 and the corresponding memory module 130 will be referred to as "local memory requests," while memory requests directed to a downstream memory module 130 will be referred to as "remote memory requests."

Local memory requests received by the downlink receiver 202 are sent through a downlink management module 210 and a controller 212 to a memory interface 214 coupled to the memory devices 148. Local memory responses are received by the memory interface 214 and sent through the controller 212 to an uplink management module 220.

Remote memory requests received by the downlink receiver 202 are sent to a downlink transmitter 216 to be sent on the downlink 134a to a downstream hub. An uplink receiver 222 coupled to the uplink 134b receives remote memory responses. The remote memory responses include an identifier portion and a data payload portion. The identifier portion, or memory response identifier, identifies the memory request to which the data payload is responsive. A packet tracker 224 is coupled to the uplink receiver. The packet tracker 224 identifies the memory response identifier. In some embodiments, when the remote memory response is sent through an uplink transmitter

226 the packet tracker 224 removes the associated memory request identifier from the packet memory 206.

A multiplexor 208 is coupled to the uplink transmitter 226, the uplink management module 220, the uplink receiver 222, and arbitration control logic 230. The multiplexor 208 couples either data from local memory responses or data from remote memory responses to the uplink transmitter 226. The choice of which type of memory response – local or remote – to couple to the transmitter 226 is determined by a control signal generated by the arbitration control logic 230. The arbitration control logic 230 is coupled to the packet memory 206, and can accordingly determine the oldest memory request in the packet memory 206. When a local request is the oldest memory request in the packet memory 206, the arbitration control logic 230 develops a control signal for the multiplexor 208 that results in the local memory response being coupled to the uplink transmitter 226 for output to the uplink 134b. When a remote request is the oldest memory request in the packet memory 206, the arbitration control logic 230 issues a control signal to the multiplexor 208 that results in the remote memory response being coupled to the uplink transmitter 226 for output to the uplink 134b. In some embodiments, remote memory responses are coupled to the uplink transmitter 226 by default. In other embodiments, local memory responses are coupled to the uplink transmitter 226 by default.

An example of the packet memory 206 is illustrated in Figures 3a-b. In the illustrated embodiment of Figure 3a, remote requests $R_0$, $R_1$, and $R_2$ were received, and the request identifiers stored in the packet memory. The local requests $L_1$ and $L_2$ were then received, followed by $R_3$, and so on. In this example, remote memory responses are forwarded as received, and the corresponding request identifier is removed from the packet memory 206. Even if the local memory response to request $L_1$ is received, if the link is in use, the local response is not sent until the request $L_1$ is the oldest in the packet memory 206, as illustrated in the example shown in Figure 3b, where responses associated with requests $R_0$, $R_1$, and $R_2$have been sent.

In the preceding description, certain details were set forth to provide a sufficient understanding of the present invention. One skilled in the art will appreciate,

however, that the invention may be practiced without these particular details. Furthermore, one skilled in the art will appreciate that the example embodiments described above do not limit the scope of the present invention, and will also understand that various equivalent embodiments or combinations of the disclosed example embodiments are within the scope of the present invention. Illustrative examples set forth above are intended only to further illustrate certain details of the various embodiments, and should not be interpreted as limiting the scope of the present invention. Also, in the description above the operation of well known components has not been shown or described in detail to avoid unnecessarily obscuring the present invention. Finally, the invention is to be limited only by the appended claims, and is not limited to the described examples or embodiments of the invention.